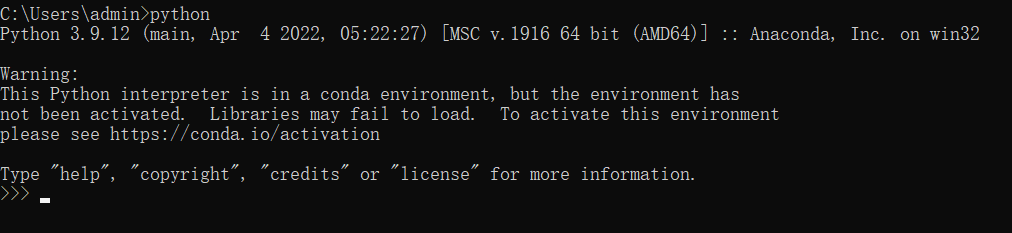
# 图像处理课堂练习（一）

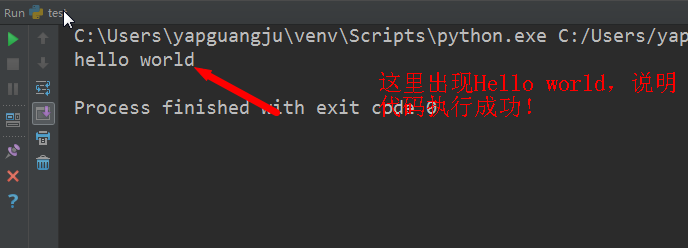
一、基本环境配置

（本节课程推荐使用python）

1. python环境搭建（下载网址：<https://www.python.org/downloads/windows/）>



1. pycharm安装及python解释器的搭建（使用pycharm运行基础python程序）。



1. （选做）安装anaconda，并使用虚拟环境成功运行python程序。
2. 基础图像处理
3. 使用Pillow、opencv等图像处理库进行图像的基本操作：
4. 图像读取与显示；
5. 图像裁剪及翻转；
6. 操作像素（修改指定区域像素值）；

④缩放与黏贴图像；

例：

1. （选做）图像处理进阶：
2. 将RGB图像转为灰度图；
3. 得到RGB图像的颜色通道直方图（hist函数）；
4. 对图像进行阈值分割；
5. 使用不同算子对图像进行边缘提取；

# 图像处理课堂练习（二）

一、基本环境配置

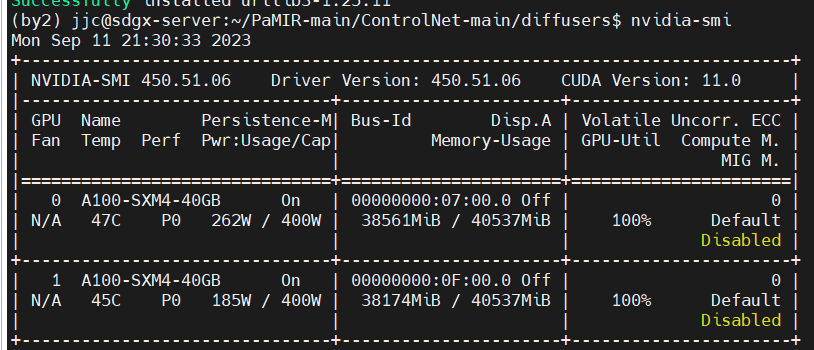
（本节课程推荐使用python）

1. （选做）安装anaconda，并使用虚拟环境成功运行python程序。

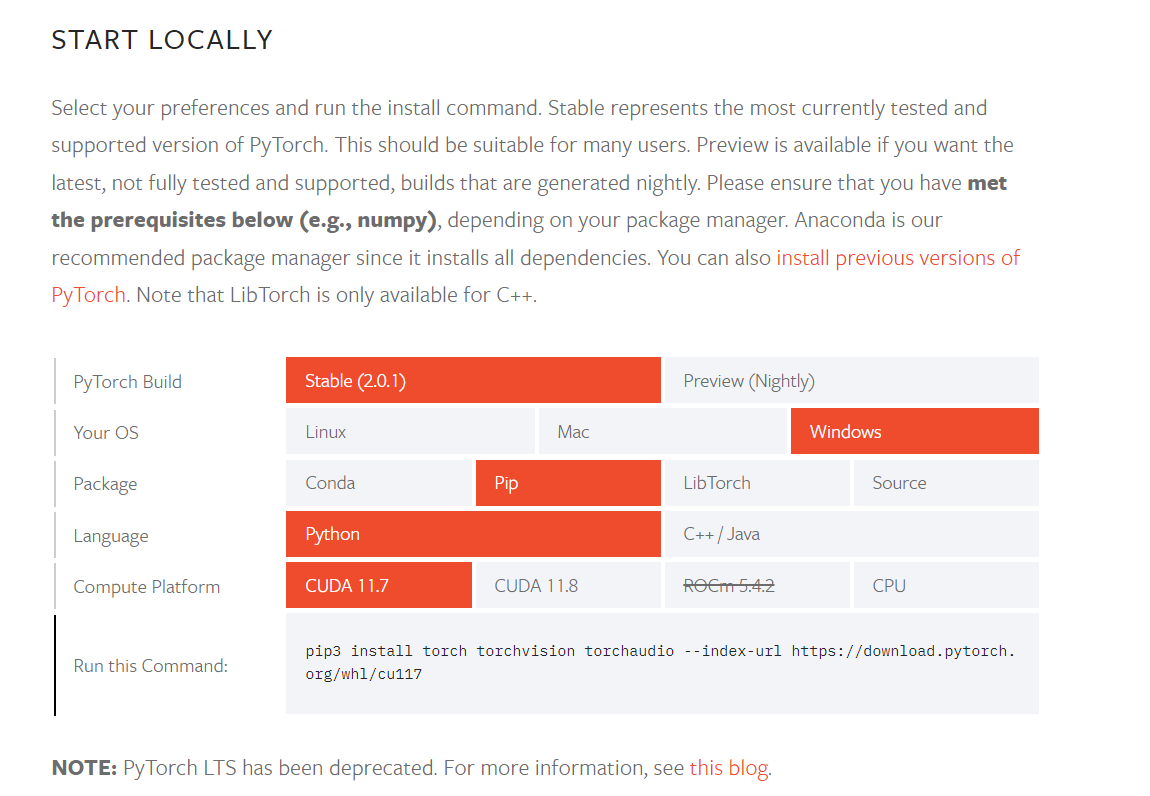
①安装CUDA和cuDNN（选做）

②配置一个虚拟环境，并根据自己电脑配置，安装版本匹配的pytorch 和 cuda工具包。

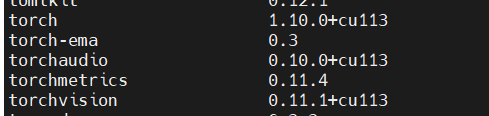
输入nvidia-smi命令，查看设备支持最大CUDA版本（安装版本必须小于等于支持最大版本）



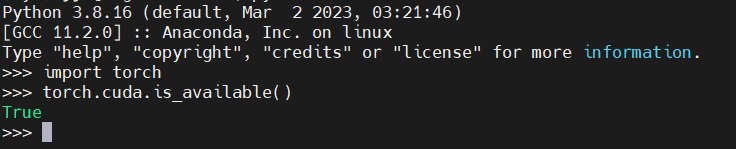
根据官网教程下载正确版本pytorch (https://pytorch.org/get-started/locally/#windows-package-manager)



输入pip list （可以查看所安装的所有库）



验证cuda是否可用



1. 手写数字识别基础（深度学习入门操作）

目的：提前熟悉深度学习实践操作，更好理解上课内容。

以下提供代码

1. [mnist数据集详解以及可视化处理（pytorch）](https://zhuanlan.zhihu.com/p/306399851)

参考代码：

1. **from** torchvision **import** datasets, transforms
3. # #下载测试集
4. # train\_dataset = datasets.MNIST('./data', train=True,
5. #
6. #                                 download=True)
7. # test\_dataset =  datasets.MNIST('./data', train=False,
8. #
9. #                                 download=True)
11. **import** os
12. os.environ['KMP\_DUPLICATE\_LIB\_OK']='True'
13. **from** skimage **import** io
14. **import** torchvision.datasets.mnist as mnist
16. root="./data/MNIST/raw"
17. train\_set = (
18. mnist.read\_image\_file(os.path.join(root, 'train-images-idx3-ubyte')),
19. mnist.read\_label\_file(os.path.join(root, 'train-labels-idx1-ubyte'))
20. )
21. test\_set = (
22. mnist.read\_image\_file(os.path.join(root, 't10k-images-idx3-ubyte')),
23. mnist.read\_label\_file(os.path.join(root, 't10k-labels-idx1-ubyte'))
24. )
25. **print**("training set :",train\_set[0].size())
26. **print**("test set :",test\_set[0].size())
28. **def** convert\_to\_img(train=True):
29. **if**(train):
30. f=open(root+'train.txt','w')
31. data\_path=root+'/train/'
32. **if**(**not** os.path.exists(data\_path)):
33. os.makedirs(data\_path)
34. **for** i, (img,label) **in** enumerate(zip(train\_set[0],train\_set[1])):
35. img\_path=data\_path+str(i)+'.jpg'
36. io.imsave(img\_path,img.numpy())
37. f.write(img\_path+' '+str(label)+'\n')
38. f.close()
39. **else**:
40. f = open(root + 'test.txt', 'w')
41. data\_path = root + '/test/'
42. **if** (**not** os.path.exists(data\_path)):
43. os.makedirs(data\_path)
44. **for** i, (img,label) **in** enumerate(zip(test\_set[0],test\_set[1])):
45. img\_path = data\_path+ str(i) + '.jpg'
46. io.imsave(img\_path, img.numpy())
47. f.write(img\_path + ' ' + str(label) + '\n')
48. f.close()
50. convert\_to\_img(True)#转换训练集
51. convert\_to\_img(False)#转换测试集

2、搭建简单的神经网络

3、定义损失函数和优化器并训练

4、趣味实验，保存训练权重，自己手写数字并拍照，将图片输入网络，查看是否检测正确。

4、选做：

①不使用minst库的数据集加载函数，请自己实现通过图片和标签文件将数据输入到神经网络并训练。

②自己编写图像裁剪，翻转等预处理操作对图像进行增广（扩大数据集），查看准确率有何变化。

③自己编写卷积函数，激活函数，并训练网络。